

Scaling up a Tangible User Interface for Education

MASTER PROJECT (Spring 2013)
Computer Science

Author:

Claude BOSSY

Supervisors:

Sébastien CUENDET

Prof. Pierre DILLENBOURG

Delivery date: 21st of June 2013

CONTENTS

1	INTRODUCTION	1
2	RELATED WORK AND BACKGROUND	3
2.1	Tangible User Interfaces	3
2.2	Learning with TUIs	4
2.3	Context	5
2.4	The TinkerLamp and TapaCarp	7
2.4.1	The TinkerLamp	7
2.4.2	TapaCarp	7
2.5	Scalability of TUIs	9
3	DESIGNING ONLINE TAPACARP	11
3.1	System description	11
3.2	Tracking	11
3.2.1	Comparison Methodology	13
3.2.2	Results	14
3.3	Rendering	15
3.4	E-Learning Platform	17
3.5	Comparison with the regular TapaCarp	18
3.6	Pilot study	20
3.6.1	Improvements	22
4	USER STUDY	25
4.1	Challenges	25
4.1.1	Security issues	26
4.2	Study set-up	26
4.3	Results	27
4.3.1	Overview of the visitors	27
4.3.2	Subscribers and the navigation pattern	29
4.3.3	Contestants results	31
4.4	Discussion	38
5	CONCLUSIONS	41
5.1	Conclusions and future work	41
A	QUESTIONNAIRE FOR SUBSCRIBERS	44
B	QUESTIONNAIRE FOR CONTESTANTS	47
	BIBLIOGRAPHY	53

INTRODUCTION

The Computer-Human Interaction in Learning and Instruction (CHILI) laboratory at EPFL has done different research on Tangible User Interfaces (TUIs). It has, along this research, developed several TUIs that all had an educational purpose. A camera-projector system, called the TinkerLamp, was created in this laboratory to be able to run different applications with the goal of improving some skills. The original application aimed to help logistics apprentices during their theoretical formation. Another application focused on improving geometry skills for children in primary schools. The most recent application, TapaCarp, was developed to help carpenter apprentices develop their spatial skills. Several empirical studies showed that TapaCarp could help to improve the spatial skills of its user.

However, despite these positive results, the systems developed for the TinkerLamp have not been adopted massively by schools. There are many reasons why this is the case, including the cost of the TinkerLamp, its heavy weight, maintenance issues and its specific applications. The goal of this work was to create an alternative solution to scale up this TUI via an online application. We will see what solution was found and how well did it perform.

In Chapter 2, a brief overview of TUIs will be presented and their possible impact on learning. Then there will be a presentation of the TinkerLamp and TapaCarp, which form the system we wanted to scale up.

Then in Chapter 3, we will present the main contribution of this work, i.e. the scalable version of TapaCarp that we developed. We will describe the different components that were implemented, as well as which technologies we used and why and the issues we encountered during the development of the project and how we solved them.

Chapter 4 will present the study that we did with our project. We will show how the study was done and the different results we obtained. Then we will discuss on the positive and negative outcome of this study and what can be done from there.

Finally in Chapter 5, we will summarize what worked well and what did not. We will also talk about the future of this project, if it worth continuing with it and if yes, what can be the next steps for this project.

RELATED WORK AND BACKGROUND

2.1 TANGIBLE USER INTERFACES

The first idea of Tangible User Interfaces (TUIs) was published in 1995 by Fitzmaurice et al. [14] but the authors were using the term Graspable User Interfaces. They were using little bricks to interact with an application. In 1997, Ishii and Ullmer [17] first introduced the notion of Tangible User Interfaces in their work on Tangible Bits. They used the word "tangible" and not "graspable" to emphasize that the user's physical environment can be the interface. After those two works, many researchers have worked on the topic. Because of the various TUIs developed, Ullmer and Ishii published in 2005 a classification of TUIs [28]. They separated TUIs into three different kinds: Interactive Surfaces, Constructive Assembly and Token+Constraint.

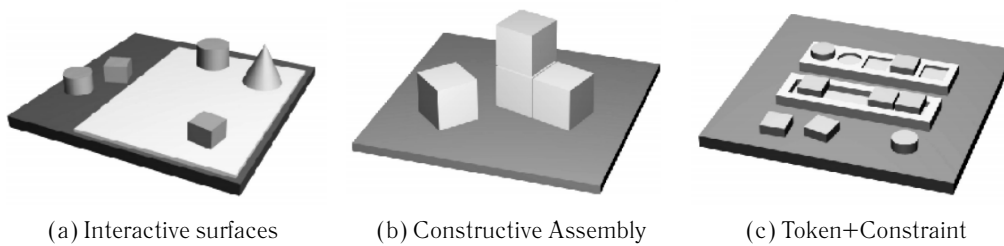


Figure 2.1: Illustration of the classification of TUIs by Ullmer and Ishii [28].

With Interactive Surfaces the user moves physical objects on a surface and the system interprets the placement of the objects. There are many such TUIs, e.g. the Sandscape [16] where users can interact with sand in a sandbox. The users can then see many informations related to the current relief modelled by the sand. There is also d-touch developed by Costanza et al. [7] which is a low-cost tangible user interface that users can build at home. Users move objects on a given work area to play some drum rhythms.

Constructive Assembly systems have objects that can be connected to each other and will give informations to the system. Connected objects can give an execution order to the system. Tern [15], for example, has instruction blocks that can be assembled to give order to a robot. It can allow children to be introduced with the basic principles of programming. Anderson et al. [3] had a different approach, in which the user can build a house out of bricks (similar to the Lego bricks). The system will then analyze the resulting 3D shape, trying to identify house properties like the roof, door or windows and finally render it with 3D graphics.

Token+Constraint systems use physical objects that have a limited range of action. For example, a tangible slider can be used in such a system, where an object can be moved horizontally or

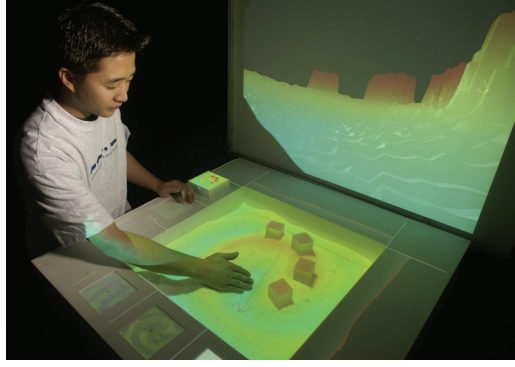
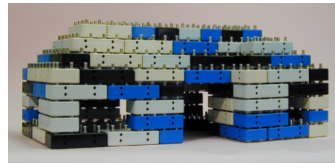


Figure 2.2: Illustration of the Sandscape [16].



(a) Block structure



(b) Interpreted 3D model

Figure 2.3: Illustration of Anderson et al. work [3].

vertically within a given range. The Tangible Query Interfaces [28] use wheels to change some parameters for querying a relational database and then output different visualizations of data.

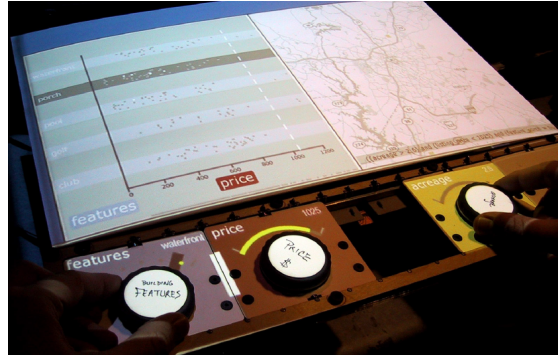


Figure 2.4: Illustration of the Tangible Query Interfaces [28].

2.2 LEARNING WITH TUIS

Shaer and Hornecker [25] listed many domains of application for TUIs such as information visualization, tangible programming and entertainment. Their list is not exhaustive and some TUIs can target more than one domain, e.g. aforementioned Tern is a tangible programming TUI, an entertainment TUI and also a learning TUI. The TUI presented in Section 2.4 is focusing on the learning application domain. There are several reasons why a TUI can be helpful for learning purposes, and they will be presented in this section.

Learning with physical objects is a feature offered by TUIs but it is not specific to it. Physical objects as a means for learning is often used with children because the younger they are, the more limited their ability of abstraction is [26]. Children often grow up playing with a shape sorting toy, where they have to put an object in the right hole. This is a typical example of a learning method with tangible objects. Researchers [24, 4, 6] have shown that manipulating physical objects help children to learn. The explorative use of toys push them to be more creative. But some other researchers [21, 18, 27, 29] raised doubts about the learning potential of the manipulation of objects. They argued that some children might be more entertained than really learning. So the fact that TUIs for learning might benefit from the manipulation of physical objects is not clear and may depend on the application.

Compared to Graphical User Interfaces, researchers showed that TUIs can increase usability [20, 13]. For some applications it can be faster and more efficient to interact with physical objects. Moreover there are fewer levels of abstraction so the users time to adapt is shortened and they can start quickly the tasks they are intended to do. For learning purposes, easy and fast manipulations of objects can decrease the learning ability [23, 12]. Developers should keep that in mind when designing their application that still gives time to the users to think why they should move an object in a given position. Random guesses are counterproductive for learning.

With some TUIs, users can see their physical objects but also one or several virtual representations of them. Zhang and Norman [30] reported several properties of external representations for cognitive tasks such as: providing memory aids, providing more direct information (less interpretation needed) and anchoring and structuring cognitive behaviour. According to research [1, 19, 22], multiple external representations can help for learning and problem solving application. Ainsworth [2] published a “conceptual framework for considering learning with multiple representations” and explained that multiple representations need to be well used to have a positive impact on learning.

Having physical objects, a better usability and multiple external representations can all improve the learning process of TUIs users. But the TUI needs to be well designed because with a bad use of each of those three properties, a TUI can also be useless for learning purposes.

2.3 CONTEXT

The following work is part of a project to develop a learning application for carpenters during their formation. In Switzerland, the mandatory education stops after middle school, teenagers, usually aged 15, can either go to high school or start a vocational education and training. During this vocational training, apprentices work directly with a company during four days a week and one day a week they go to a school to learn the theory behind their trade. One of the goals a carpenter apprentice has to achieve is the ability to read, understand and be able to project plans given by architects or engineers¹. Those plans are orthographic projections of a 3D model. Understanding orthographic projections is not trivial, it can take time, it requires good spatial skills and is typically taught during their one day at school through descriptive geometry classes.

¹ <http://www.orientation.ch/dyn/1109.aspx?id=39>

An orthographic projection is a view of a 3D model from a given point of view and unlike perspective projections it uses parallel projections, i.e. an element that is further away is not drawn smaller. There is an illustration in Figure 2.5 with a cube viewed from the same position but the Figure 2.5b shows a perspective view and the Figure 2.5a is an orthographic view. Orthographic views are more difficult to understand because this is not how the human eyes see things. It is harder to internally visualize a 3D object represented like this because with only one orthographic projection, there is less depth information. The stippled line in Figure 2.5a means this line is behind the front face of the cube but it would have been drawn the same way if it was even further away from the camera.

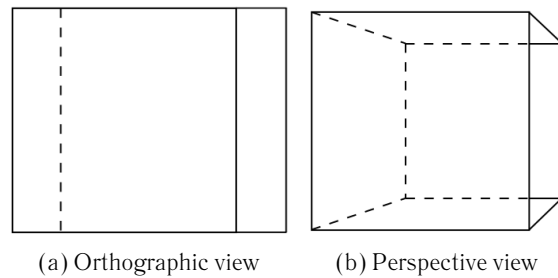


Figure 2.5: Two different projections of a cube.

Carpenters have to deal daily with three different orthographic projections, the front view, the side view and the top view. In Figure 2.6, there is an example of those three views. There are also different colors to help understand those views. If someone is looking at the object like in the perspective view in Figure 2.6d, the front view is as if the camera is facing the magenta wall, the side view is as if the camera is facing the cyan wall and for the top view the camera is facing the yellow ground. The plans also show objects as if they were transparent and the stippled lines means that those lines would not be visible if the objects were completely opaque.

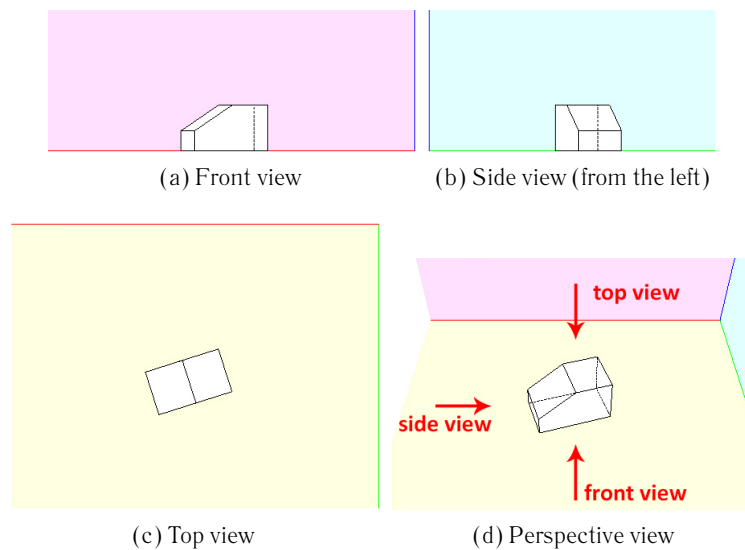


Figure 2.6: Orthographic projections

The apprentices at school are learning how to construct those views from a real object and how the three views are related to each others. For example, they should be able to draw the top view with only the front and side views. Manually drawing this, takes a lot of time because they need to draw construction lines, as shown in Figure 2.7, and find the intersection points from the front and side views to be able to draw the final shape of the top view. This helps the carpenters improve their spatial skills but the shapes they have to draw are more complex than the one below and thus takes even more time.

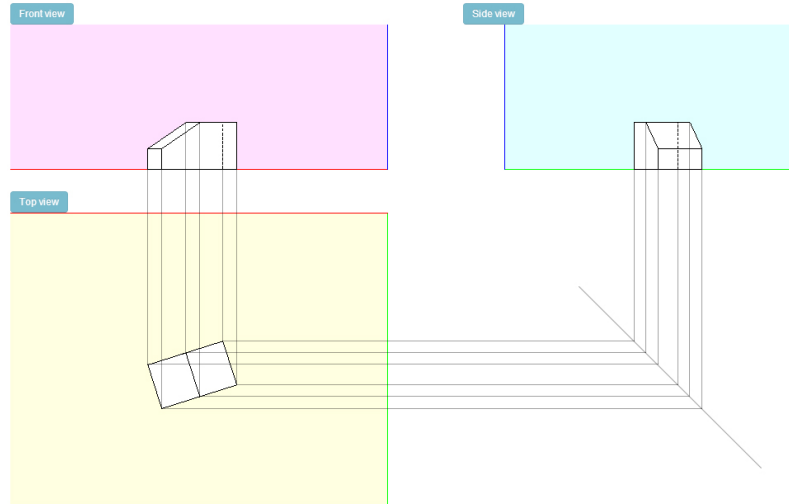


Figure 2.7: Construction lines to draw the top view from the front and side views.

2.4 THE TINKERLAMP AND TAPACARP

2.4.1 *The TinkerLamp*

The CHILI laboratory at EPFL has developed a machine called the TinkerLamp with a first application for apprentices working in logistics. The TinkerLamp (shown in Figure 2.8) is composed of a computer, a projector and a camera. With the use of a mirror it can project a graphical interface on a tabletop and record images of this interface with the camera.

The application for logistics, shown in Figure 2.9, provides small-scale shelves to interact with the system. The shelves have a fiducial marker on top of them so the camera can track their position. The goal is to build a model of a warehouse and then the user can run simulations to see how effective the placement of the shelves is. It is also using a paper-based interface, i.e. the user receives paper sheets that allow him to start activities and provide questions the TUI can help to answer. Zufferey et al. [31] showed that the system has a good usability and also that it is useful in a real context (professional school for logistics apprentices).

2.4.2 *TapaCarp*

Cuendet et al. [10] noticed a gap between the carpenter apprentices' work in companies and their theoretical courses at school during their vocational education. They realized that the ap-



Figure 2.8: The TinkerLamp.

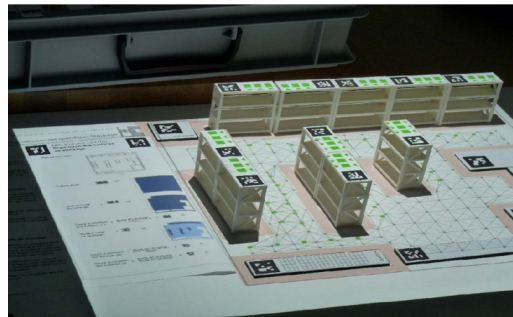


Figure 2.9: The application for apprentices in logistics.

prentices had a lot of drawing lessons even though carpenters draw rarely during their practical work. Those drawing lessons are useful anyway because this is how they develop their spatial skills. But Cuendet et al. saw there an opportunity for improvement and with the positive results of Zufferey et al. [31] they developed the TapaCarp system for the TinkerLamp to develop a way of improving a user's spatial skills.

TapaCarp contains many activities but the default activity displays the front, side, top and perspective views of an object. The user can manipulate the object and the views are dynamically updated according to the new object's position. The user can also use cards to display different kinds of information, in Figure 2.10, the card makes the system draw the construction lines that we explained in the previous Section.

There is one activity involving two users. The first user can place objects as he wishes, then asks the system to save the position. The second user has then to place back the objects knowing only the front and side views. The top view is updated according to the second user's object manipulations and the user can use the card to render the construction lines to help him find the solution. If proved efficient, this is a more entertaining and faster alternative to understand the relations between the views and develop spatial skills than the current solution of drawing all

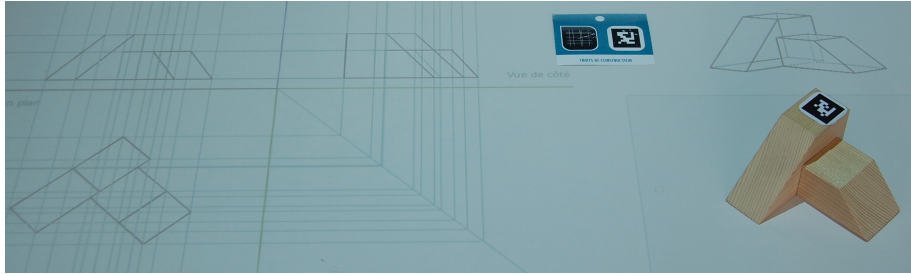


Figure 2.10: Construction lines with TapaCarp.

the lines. This avoids the mechanical part of drawing which is useless for the learning process of spatial skills.

In Section 2.2, we have seen that TUIs can be a helpful mean for learning, if used correctly, and TapaCarp has been carefully designed to benefit, as much as possible, from the learning abilities of TUIs. Recently Cuendet et al. [8] showed the impact of TapaCarp for learning spatial skills with carpenter apprentices. The spatial skills were significantly improved with the use of TapaCarp. In the same paper, they also showed the difference between using real physical objects and tokens. Tokens were just tangible bricks to interact with the system, as opposed to the real physical objects that have the same shape as their virtual representation projected on the tabletop. The latter did not improve the learning outcomes of the users but improved their performances through the various tasks they were facing.

2.5 SCALABILITY OF TUIs

Research on TUIs has grown over the last two decades but most of the produced TUIs are just displayed in museums or only used for studies. Not many TUIs are used regularly in real context and TapaCarp also is facing this issue, it has not been adopted in school yet, except for Cuendet's studies [8, 9, 11].

TUIs are usually designed for specific purposes. For example, TapaCarp's goal is to help carpenter apprentices through their theoretical formation. It is focused on 2D and 3D geometry, so it can aim for some other potential users like architects or mechanical engineers. But TUIs are usually designed for a niche market, at least at first because of the long development time and high cost.

Most of the TUIs need a dedicated software, a computer to run on, the tangible objects and often a dedicated piece of hardware. Since TUIs are based on specific applications, the required hardware often has to be sold with the software. This makes TUIs expensive and thus hard to sell and spread. The TinkerLamp, in particular, requires expensive hardware (computer, projector and camera) but there is also the designing and manufacturing of the TinkerLamp. It is difficult to sell those machines to the schools because it is hard to prove to the schools that the benefits of using this lamp is worth the prize.

The TinkerLamp is a heavy machine, it weighs around 15kg. So shipping TinkerLamps is difficult or again expensive. This narrows a lot the potential customers if it cannot be sold world-wide. Moreover, if for example there is a problem with the computer, this machine is not easily repairable for the users. Maintenance would also be problematic if customers are spread world-wide.

The goal of the work presented here is to tackle those challenges and find a more viable solution, i.e. a cheaper and lighter version of this system. As explained before, TapaCarp can improve the learning process of the carpenters. In this work, we will not try to find other target users but we want to find an alternative to the expensive TinkerLamp. We need a low-cost system that can be easily spread and thus be more appealing to the professional schools.

The solution presented in this work is to create a web application based on HTML5 standards. The projector is replaced by a screen, the camera by a webcam and we assumed the users own a computer connected to the internet. This is a reasonable assumption for developed countries, e.g. in 2012, 76% of households in the European Union has an internet access². The tangible objects could be printed and build out of papers. So the users with a computer, an internet connection, a webcam and a printer would not need to spend any money to access this learning TUI.

In Switzerland, professional schools already own computers and they would probably only need to buy webcams to fit the hardware requirements. There are four reasons why we decided on a web application. First, JavaScript has been more and more used lately because of the growing number of open source libraries and that it runs on the client side so it can save the server from many computations and can render 3D graphics. Second, HTML5 standards are being more and more mainstream and compatible with most of the browsers. Third, it is easier to maintain a web application even though it can be difficult to have software compatible with every operating system, internet browser or webcam. But if the user is respecting the requirements, we can find solutions to their problem wherever they are located. And the last reason is simply that having an online application is the easiest way to achieve more visibility and thus increase the scalability of TapaCarp.

² http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-SF-12-050/EN/KS-SF-12-050-EN.PDF

DESIGNING ONLINE TAPACARP

3.1 SYSTEM DESCRIPTION

The goal of the system is to show several representations of real 3D objects that the user can manipulate. Figure 3.1 illustrates the system running. There are tangible objects on the tabletop and on the screen there are several virtual representations of the tangible objects. No extra software on the computer is required. All the user needs is an internet browser, a webcam and printed material. The system needs to be able to get the position and rotation of the objects, to draw various representations of the real objects. This highlights the requirement for the first two components of the system: a tracking system and a rendering engine. Because it is online and we want the user to have a personalized experience, the system relies on a web-platform. This chapter presents the components of the system.

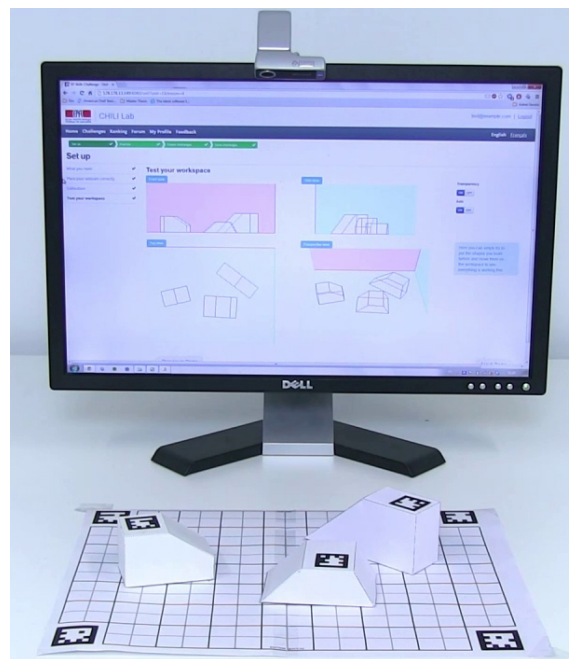


Figure 3.1: Illustration of the system

3.2 TRACKING

To represent the objects virtually on a plan, we needed to define this plan in the real world too. This is the purpose of the grid (hereafter refer to as "workspace") that you can see in Figure 3.1. If the user puts an object in the middle of this grid, it should be in the middle of the 3D virtual representation as well. On the camera image, this grid is distorted, with the distortion depending

on the camera characteristics and position. However, with an homography we can map the coordinate system of the webcam (pixel coordinates) to the grid coordinate system (real world coordinates in millimeters). All that is needed for an homography is four points that can be seen in both systems. That is why we needed a tracking system to get the position of the workspace and the objects in the webcam image.

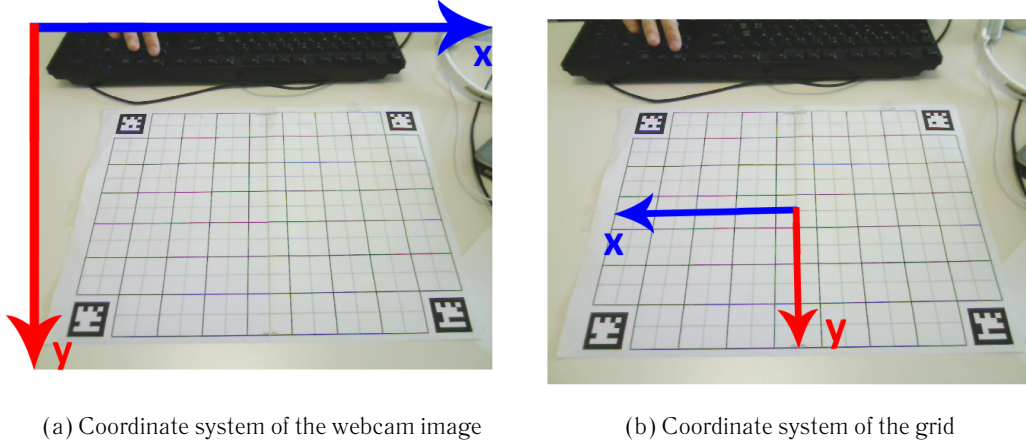


Figure 3.2: Mapping of coordinate systems

To get the position and orientation of the objects and the workspace, we use fiducial markers. Fiducial markers are matrices of black and white squares inside a black square. These matrices can be seen as a bit string where a black square means 1 and a white square 0. Examples of markers are shown in Figure 3.3. A marker has a border that allows it to be detected by the algorithm of the fiducial marker detector. This border is usually of the size of one or two squares. Inside this black border is the information matrix. For example in Figure 3.3a, there is a 5×5 matrix inside a border of one square and in Figure 3.3c there is a 6×6 matrix inside a 2-square border. The inner matrix serves two purposes. Error checking, to make sure this is an expected marker and not a random square; and encoding information to differentiate markers and associate a value. In Chilitags¹, only 10 cells in the matrix are used to encode information, which means it can detect 2^{10} different markers, and 26 cells are used for error checking.

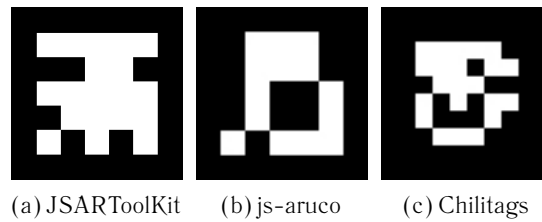


Figure 3.3: Fiducial markers for different libraries

¹ <https://github.com/chili-epfl/chilitags>

A fiducial marker library is a set of computer code that uses computer vision to detect those markers on images. We found two such JavaScript libraries: js-aruco² and JSARToolKit³. We tested both to see which one best suited our purposes. In our project we use markers for two purposes: to know the position of the objects the user is moving, and also to know where the webcam has been placed by the user. We therefore need a fiducial marker library. Another constraint is to allow the detection of small markers, because the markers will be put on objects whose size can vary. The size of the marker in the camera image varies with the camera distance. And if a camera is not directly in front of a marker, the shape of the marker will look more like a trapezoid than a square and is thus harder to detect. We want the library that allows the biggest range of distance and angle from the camera to the marker.

Some differences can already be noticed before any empirical evaluations of these two libraries. JSARToolKit is a JavaScript version of ARToolKit, which is a widely spread library in C and C++. It is meant to be used for augmented reality applications and needed to be adapted for our purpose to be able to easily get the pixel coordinates of detected markers. On the other hand, js-aruco is much simpler (about 10 times fewer lines of code). It only implements the functions from OpenCV needed to be able to detect markers and to return their pixel coordinates. So, at first sight, js-aruco looks like it fits our needs better. However, before choosing between these two libraries we want to compare their performances.

3.2.1 *Comparison Methodology*

The goal is to see how the libraries perform with different camera positions, i.e. at different heights and angles. The height is the distance between the marker and the camera when the webcam is vertically align with the marker. As illustrated in Figure 3.4 the angle is the one between the marker plan (e.g. the tabletop on which it is placed) and the line from the camera position to the center of the marker. The best position is when the camera is directly above the marker (so 90°) because there is little image distortion from the perspective and therefore appear as squares. The larger the angle, the more squares become trapezoids and the harder the detection is.

With HTML5 it is easy to access a webcam from the browser but for now it can only stream with a resolution up to 640×480 pixels. This means that having a brand new webcam or a cheap one does not matter as long as the webcam resolution is not below 640×480. But there is a parameter that matters: the field of view. Since all webcams will stream to the same resolution, if we put two different webcams at the same position, the webcam with the biggest field of view will see markers smaller. So to determine the minimal size of the markers to use in our comparison, we used the webcam with the biggest field of view. It was the Logitech C920 HD with a field of view of 78° against 48° for our cheapest webcam. We put it at a height of 30cm, which is the lowest height to avoid the camera interfering with the user hands manipulations. The smaller markers that could still be well detected at this height had a side size of 20mm. So this marker size was used for the comparison between the two libraries.

² <https://code.google.com/p/js-aruco/>

³ <https://github.com/kig/JSARToolKit>

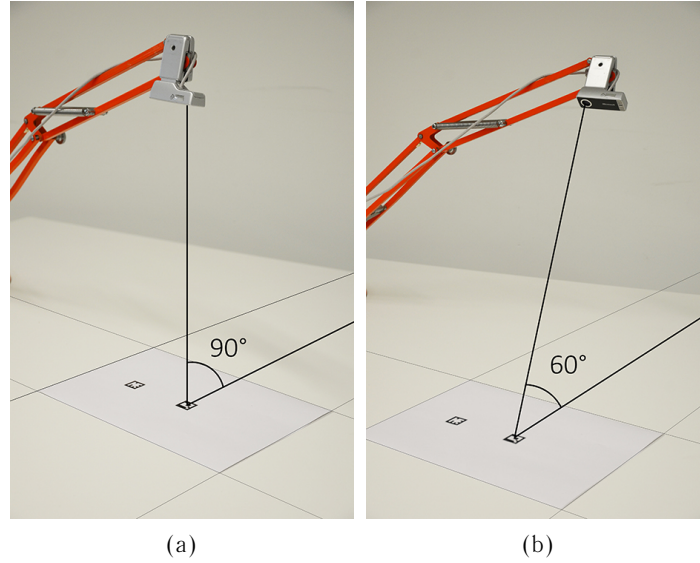


Figure 3.4: Illustration of camera angle

Since the goal of this work is scalability, to compare the two libraries we used a Sweex WC150 USB webcam, which is a cheap camera (18,57EUR on amazon.fr, 23rd of May 2013). We first tested the different heights from 30cm to 55cm with a 5cm step. The lower bound has been explained before and the higher bound has been chosen because the higher the webcam is, the smaller markers are and the less precision we get from pixel coordinates. For each height we checked on 100 frames in how many frames the marker was detected.

Then we also checked how the markers were detected if the webcam is not directly above but with some angle. The angle was computed given the horizontal and vertical distances from the camera to the marker. The libraries have been tested from 45° to 65° with a 5° step (as a reminder 90° means the camera is vertically aligned with the marker). So the performance should decrease as the angle decreases too. The lower and higher bounds were simply chosen by moving the camera and observing from where the marker detection seemed to be less stable for either of the two libraries. To change the angle we simply changed the horizontal distance. It means that the smaller the angle, the further the camera is from the camera. This was done for practical reasons: it was difficult to get a stable position for the webcam. But both libraries were tested the same way.

3.2.2 Results

As shown in Figure 3.5a, for all heights the JSARToolKit library detected the marker (98 frames out of 100 for a 55cm height). For js-aruco the detection rate was similar to that of JSARToolKit up to a height of 45 cm. But then it dropped to 75% of frames where the marker is visible for a height of 50cm and even no detection of the marker for a height of 55cm.

Figure 3.5b shows the results of the comparison for the different angles. As can be observed, once again JSARToolKit performs better than js-aruco. Using JSARToolKit, the camera can

have a 50° angle with the marker plan and still be well detected (on more than 90% of the frames, the marker was detected), while using js-aruco the detection of markers stopped at 60° .

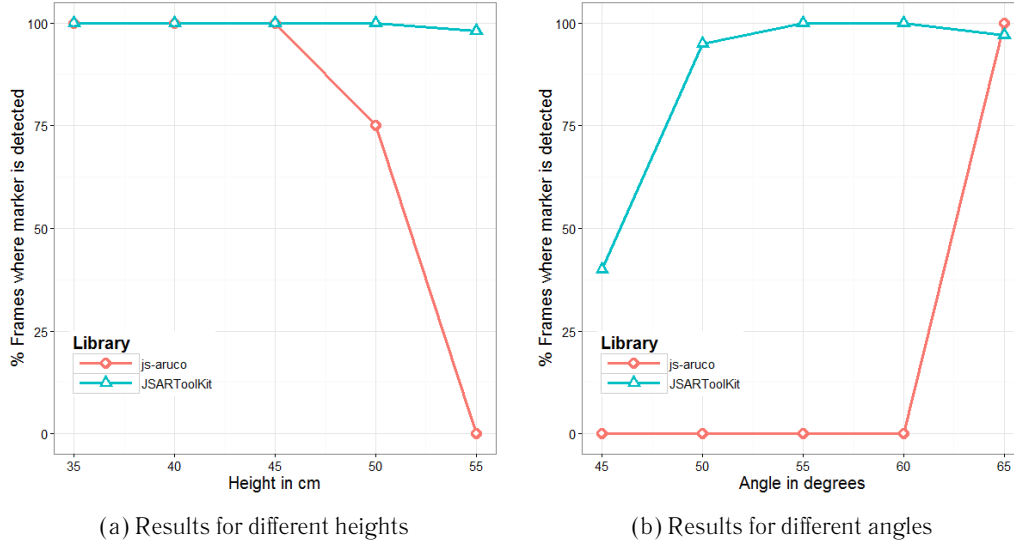


Figure 3.5: Results with 20mm markers

It was difficult to measure precisely distances and angles with only a ruler and a set square, and it is possible that there was an approximation of a couple of degrees or centimeters. But since the libraries behaved very differently, those results were helpful anyway. After realizing how well JSARToolKit's performances were, compared to js-aruco, we tried a test more related to our system's configurations. The tangible interface among other things is composed of a workspace, which is the size of two A4 papers with a marker on each corner. An important part of the application is the ability to retrieve the camera position using the markers of the workspace. The system needs to detect all four marker corners to compute the position of the camera. For both libraries, we tried to see the extreme heights and angles where the webcam was still able to detect the four markers. The markers on the workspace have a 33mm side size, which is a good compromise on the workspace. It is not too big and is still well detected if the webcam is on top of a computer screen, which is the most common position of a webcam.

The results of this more practical experiment can be seen in Table 3.1. JSARToolKit, as expected, is again performing better. We can put the webcam 23cm higher i.e. nearly 30% higher than with js-aruco's limitation. Regarding the camera angle, js-aruco allows a range of 20° (from 90° to 70°) while JSARToolKit's range can get up to 50° , which corresponds to an increase of 150% from js-aruco's performance. All these results made us choose JSARToolKit for our system.

Library	Max. height	Max. angle
js-aruco	87.5cm	70°
JSARToolKit	113.5cm	40°

Table 3.1: Extreme values where the workspace is still detected (heights and angles between the camera and the workspace's center).

3.3 RENDERING

The system needs a rendering engine because it will draw non trivial objects such as 3D objects viewed from different points of view. To render 3D graphics on a web page, there exists a JavaScript API called WebGL. WebGL is accessing the GPU using the OpenGL drivers of a graphic card. OpenGL is a widely used 2D and 3D graphics API, it is cross-language and multi-platform. With WebGL there is no need for plug-ins, it uses the web standards but needs the browser to access the GPU. Google, Apple and Mozilla are members of the WebGL Working Group⁴, and their browser, Chrome, Firefox and Safari respectively, therefore support WebGL. On the other hand, Internet Explorer does not support WebGL yet. It is rumored to add it in version 11, which beta version should be released during the summer 2013. Fortunately, the supported browsers represent more than 80% of the market share as shown in Table 3.2.

2013	Internet Explorer	Firefox	Chrome	Safari	Others
April	12.7%	27.9%	52.7%	4.0%	2.7%
March	13.0%	28.5%	51.7%	4.1%	2.7%
February	13.5%	29.6%	50.0%	4.1%	2.8%
January	14.3%	30.2%	48.4%	4.2%	2.9%

Table 3.2: Internet browsers market share in early 2013 according to www.w3schools.com

Rendering 3D graphics with WebGL is not easy because it is a very verbose and low-level language. It is similar to OpenGL, since it is actually based on OpenGL ES 2.0, a subset of OpenGL designed for embedded systems. To draw simple 3D objects like a cube it can take a lot of code. And for our application we want to draw non trivial object. So it is good to implement classes or functions to do this more easily. Fortunately, there is the three.js library which does it for us. This library can save us a lot of time and is easy to use. It is widely used by developers who want to do 3D rendering on the web. Its developer is very active and always trying to improve it or correct bugs, and there is a large community of users.

⁴ <http://www.khronos.org/webgl/>

One important rendering feature that we needed was to be able to draw stippled lines. This is typically not easy to do in WebGL, but with three.js it can be done easily. However we still encountered a problem with three.js, where edges occluded by another object (and should therefore be dashed) were not displayed at all (see Figure 3.6). So, as for the JSARToolKit library, we had to modify slightly the three.js library to make it suit our needs.

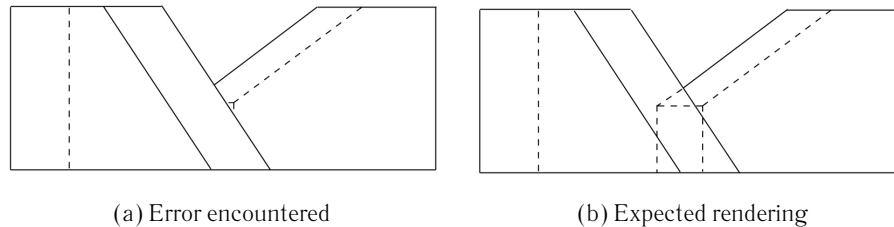


Figure 3.6: Problem with three.js

3.4 E-LEARNING PLATFORM

Each user should have its own experience. So the system needs to differentiate the users and keep track of their progress. Since it is not trivial to use and create its own tangible user interface, the system has been designed as an online course. The users follow a step by step tutorial to learn how to use the system and then also learn some 3D geometry principles. The system relies on an e-learning platform to do that.

The e-learning platform we chose was Google Course Builder⁵. There were several reasons for that. The first reason is the set of tools that comes with it: login, HTTP Secure, user progress, forum and announcements. The second reason is Google itself: it is a powerful company, they have the possibility to improve this Course Builder and they are doing it. Ever since the first release in September 2012, they have released updates every month. Additionally with the emergence of MOOCs (Massive Open Online Courses) there are good chances Google Course Builder will last. Google is currently contacting universities to invite them for Google Course Builder workshops to motivate the use of its tool. The third reason is related to the previous one, it is the large community behind Google Course Builder. For a young software like this one, the community is already large but it is not a surprise given the popularity of its creator. The last reason was the programming language. Google Course Builder is written in python and it is a language we were familiar with, so it was faster to start using it.

Right now courses made with Google Course Builder are simple. They are divided in lessons and for each lesson the administrator (normally the teacher) can add a video of the lesson and then optionally add an evaluation for the students. The evaluations are basically forms with multiple choices questions or text inputs. The advantage for the teachers is that they can create their course with a Content Management System and keep track of the students progress with evaluations. For our purpose it was too basic so we needed to add some coding on top of the Course Builder source code. We added our tangible user interface's activities, changed the

⁵ <https://code.google.com/p/course-builder/>

graphical interface, added more informations in the database, added the possibility to change the language (english or french), etc.

3.5 COMPARISON WITH THE REGULAR TAPACARP

There are some differences between TapaCarp and its online version. First, the online version needs a workspace that the user has to print and the camera has to detect. Second, there is a beamer inside the TinkerLamp and the image is projected where the objects are. So the projection and the tangible interface are said to be co-located, as opposed to the online TapaCarp where the screen is separated from the tangible objects. Third, the TapaCarp system, unlike the online version is fully controlled. The size of the projected image, the camera specifications, the size of markers as seen by the camera and the running operating system, all this information is known in the TapaCarp system but not to the online TapaCarp.

The regular TapaCarp is showing to the user a workspace where it is supposed to move objects. This workspace is only a visual help for the user. But the one of the online version gives the system important information to render the objects correctly. This is due to the fact that online TapaCarp does not know which webcam the users have and where they will put their webcam. So we had to create this workspace with four markers in order to represent the objects virtually and accurately. As explained in Section 3.2, the information retrieved from the workspace is mandatory to the system to relate the virtual objects with the tangible ones. It is also needed in the regular TapaCarp but is known a priori and is the same for all users.

We mentioned there is a difference in the representation locations between both versions. To reuse the terms of the literature [23], regular TapaCarp uses a co-located mode of representation and online TapaCarp uses a discrete mode. In the aforementioned paper, the writers noticed differences between the two modes. But there is not a better mode than the other, it depends on the application itself. The co-located mode allows a faster interaction and enhance a more explorative use of tangibles. On the other hand, separating the tangibles and the graphical interface make it slower to interact with it. The slower usage of tangible interfaces with the discrete mode allows the user to think before performing actions. Following these claims, it may be that the discrete mode could be more useful in our settings, but it would need a dedicated study on this to be able to assess this statement. Moreover the regular TapaCarp activity with the three orthographic views separate the projected interface in two. Figure 3.7 shows this regular TapaCarp activity running. On the left, there are the three views and on the right the object the user can move. The separation is very similar to the one in online TapaCarp that was illustrated in Figure 3.1. For this activity in particular, it seems that the regular version does not offer a different experience than the online one.

With the online system we encountered several specific challenges. What are the problems with having different webcams as input? Where will the user put its webcam? Are the used technologies compatible with every operating system and internet browser? How to explain to the user how to build his own interface?

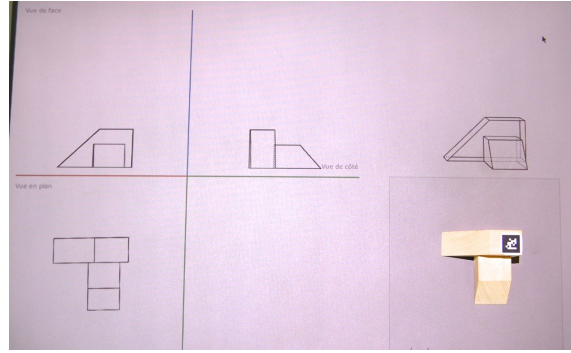


Figure 3.7: Illustration of the regular TapaCarp activity

Having different webcams did not give implementation problems, as the code works without knowing the properties of the webcam. The difference, as already mentioned, rely mainly on the field of view of the camera. The larger the field of view is, the closer to the workspace the camera needs to be. So we added feedback to the users (see Figure 3.8) to tell if the position they found for the camera was good enough or not and how to improve it. Also, some webcams do not have an autofocus so the user should be aware of that and adapt the focus manually.

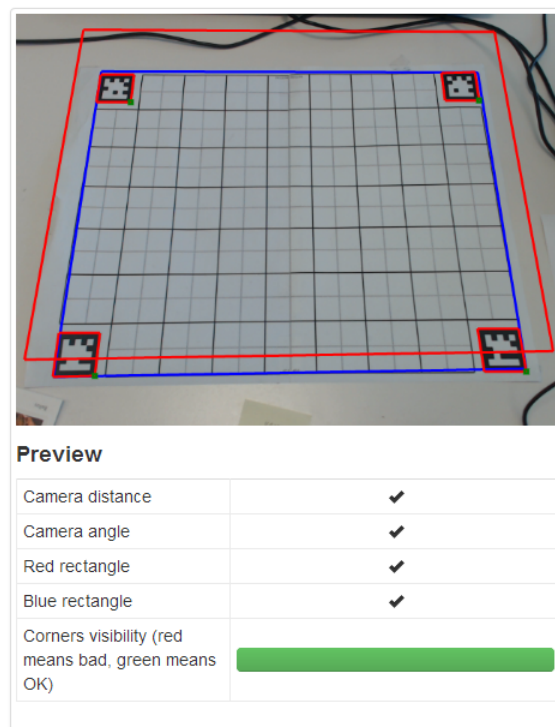


Figure 3.8: Illustration of the feedback for the webcam positioning

The internet browsers are not all compatible with the system. We have already mentioned that Internet Explorer does not support WebGL for 3D rendering. But the current version 6 of Apple's Safari is not fully compatible with our website either. It does not allow the possibility to access the webcam. We are using the `getUserMedia` API from the standard Web Real-Time

Communication (WebRTC) API which is still a work in progress of the World Wide Web Consortium (W3C). We kept this solution because it was the only one without the need of plug-ins and since it is a W3C project it will probably be adopted by other browsers soon. But now the remaining compatible browsers are Google Chrome and Mozilla Firefox which fortunately still represent 80.6% of the browsers market share (in April 2013, see Table 3.2). On the operating system point of view, online TapaCarp works fine on Mac OS X and Windows 7. It can work on Linux too but the user may encounter some difficulties due to the webcam or graphic card drivers.

Unlike in regular TapaCarp, the user does not receive the objects used for the tangible interface. We needed to find a way to make the user build its own interface. For that we had to give documents that the user would print, and then explain clearly how to build everything from the printed papers. Everything is then made out of paper (folded and glued). Handymen can also find the dimensions of the objects to cut them in wood or other materials they want. On the website we added a video and a tutorial that can guide the user on how to create and use their homemade tangible interface.

3.6 PILOT STUDY

After setting up the tracking, rendering and e-learning platform, we added a tutorial for the user to guide him through the installation and use of this tangible user interface. The tutorial is composed in four parts: installation of the interface, practicing challenges, creating challenges and solving challenges. What challenges are, will be explained in details in Chapter 4.1. For now we will consider them as exercises to understand the relation between the front view, the side view, the top view and the real world. When trying to solve a challenge, the time taken by the user to succeed is recorded. Users can then see their time and compare their performance with other users'. An overall ranking is computed to add some competitiveness between the users.

We conducted a first study with four colleagues. Some were more familiar with TapaCarp than others. The goal of the study was to test the usability of the system. This study was done in two phases. The three first users used the same first version of the system. We then implemented a second version based on the comments of the three first users. Finally the fourth user was tested on this second version of the system. The four users had to do the same task which was to follow the tutorial until the end. We gave them the paper-based interface to save them some time and we stayed by their side while they were trying to complete the tutorial. We helped them if they asked for our help or if we saw they needed guidance. We want the website to be as intuitive and self-explanatory as possible. They were not asked to comment everything they were doing but they were informed that every suggestion or comment is welcome.

The first user struggled mainly on positioning the webcam. He tried first to find a position that was working but while holding the webcam in his hand. He thought he could keep holding the camera and still use the website. Apparently it was not clear that the webcam should have a fixed position. After explaining that he could not hold the webcam throughout the whole tutorial he tried to find a place for the webcam. It was hard for him to find such a place where the webcam could detect the workspace easily. Before doing the tutorial his webcam was on top of his screen



Figure 3.9: Observation of the user's experience

but he was afraid that the webcam would fall if it would be facing down on his desk where he put the workspace. He suggested that we added some pictures to illustrate some possible webcam positions. This user was never really sure what to do, he wanted more detailed explanations. During the practice phase of the tutorial, he had some problems because there was no feedback on why his solutions were not correct.

The second user faced different problems. He had some kind of tripod with a clip on it (see Figure 3.9), so it was easier to place the camera. His main concern was how to navigate through the tutorial. He often asked where he should click. There was some issues with the rendering of grey and black lines on the grid. The grey lines were not rendered uniformly, some were darker than others. Since he was familiar with the original TapaCarp, he was looking for cards to complete some actions like "save" or "check solution". There was some jittering, i.e. the objects were constantly moving on the screen even when they were not moved in the reality. This was caused by the detection of markers that, for each frame, may detect the markers with slightly different coordinates. Sometimes he was trying to be very precise when moving his objects and the jittering frustrated him. As explained previously, the workspace needs to be detected by the webcam, but then if neither the workspace nor the webcam move, the detection of the markers on the workspace is not required since the system knows how to go between the real world referential and the camera one. Not knowing that, the user was scared to put objects on top of the workspace markers. Unlike the first user, he was familiar with the orthographic projections and did not have problems with the practice phase without feedback on his solutions.

The third user's experience was more similar to the first one. The main issue was the camera positioning. He also tried keeping the webcam in his hand. He first tried an easy position where the webcam was stable, but it was too far. The other position ideas that he tried were not usable even though the workspace was detected. For instance he tried to put the workspace on the ground with the camera on the edge of his desk. But with this position, he could not move objects and watch his screen at the same time. This condition was apparently missing in the tutorial. He did not want to read the explanations and skipped when there was too much text. He read some parts only when he was stuck during the tutorial. His previous knowledge of the TinkerLamp helped him a lot for the rest. He did not have trouble with the usability of the website.

3.6.1 *Improvements*

After the three first test users we noticed several issues:

- navigation problem, the user may get lost during the tutorial;
- the users did not understand easily how to position the webcam and were not very imaginative to find a solution;
- the practice phase was frustrating because they could not figure out what they did wrong.

To enhance the user's experience and the intuitiveness of the website several improvements were implemented. We added some pop up that shows up when the user complete some actions and guide them to the next step of the tutorial. We want to avoid the user to be lost during the tutorial. There is now more details for the explanation of the positioning of the webcam with some illustration of possible webcam positions. There is also dynamic feedback to guide the user while he is finding a place for his webcam. We added also some feedback for the practice phase, for each object that the user has to place, he can see if the object is detected and whether its position and rotation are correct. With this feedback it is easier for the user to understand his mistake.

Thanks to these improvements, this last user's experience was much better. Although she still encountered some difficulties with the webcam positioning everything went much smoother. Additionally the issue for the positioning was because the angle allowed by the tutorial's explanation was too high. Despite the lack of explanations, she understood that there was a problem even though the reason was not clear. She also showed more creativity in finding a better position. For example after putting the webcam on top of the screen, the workspace was too far so she elevated it with books. After the installation of the interface, she did not face any issues anymore.

This last observation showed that the improvements seemed to increase the usability of the website but we improved once again the tutorial for the position of the webcam. The user can now see five different dynamic kinds of feedback (see Figure 3.8): distance, angle, red rectangle, blue rectangle and workspace's marker corners visibility. So the user can see if the camera is too far or too close from the workspace and if the angle (described in section 3.2) is too low. The blue rectangle represents the contour of the workspace grid and needs to be entirely visible by the camera. The red rectangle adds needed margins around the workspace and also needs to be entirely visible on the camera image. The markers on objects have a different height than the markers on the workspace. If we put an object at an extreme border of the workspace the marker coordinates might be outside the workspace limits represented by the blue rectangle (see Figure 3.10). The red rectangle adds margin to make sure that the objects' marker can be detected as long as the objects are within the workspace. Finally the marker corners visibility represent how well the fiducial markers of the workspace are detected.

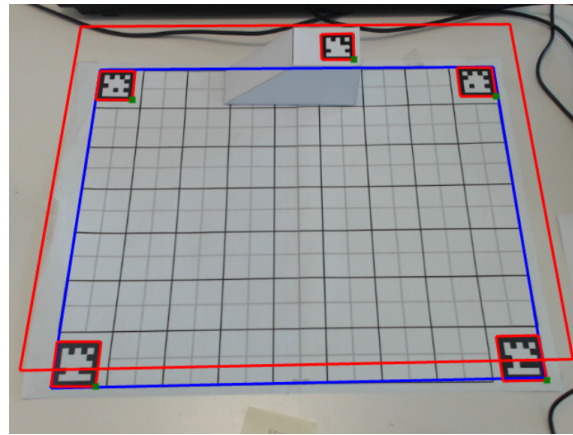


Figure 3.10: Illustration of the red and blue rectangles

The main issues we noticed were: the website navigation system, the explanations for the webcam positioning and the lack of feedback during the practice phase. It is now easier to use without any external help. We notice that explaining to the user how to place its webcam is far from trivial. So we added different dynamic feedback to inform the user if the position is good and if not, why. The navigation is now more intuitive, it is harder for the user to get lost during the tutorial. To avoid the jittering of 3D objects we implemented a stabilizer, so it will not take into account small movements. This study was really helpful. Each user can behave differently under same conditions, so it is not always simple for developers to predict what can cause trouble for the users.

USER STUDY

After the pilot study, we decided to publicly release the online version of TapaCarp. The website is separated in two parts: the tutorial and the challenges. The tutorial was improved as explained in Section 3.6 to be as self-explanatory as possible. This way we could just send an e-mail with the website URL and let the users try to use this online TUI on their own. The challenges, explained in the next Section, were created to attract more users with a contest and the possibility to win prizes for the best three ranked users.

4.1 CHALLENGES

Each user can participate to a contest after finishing the tutorial. The users have to solve so-called challenges. The user's score for a challenge depends on the time taken to complete the challenge. The ranking of users is calculated based on the total score of each user.

A challenge is an exercise where the user can see on the screen a front view and a side view. The two views represent the same positioning of objects. There are either one, two or three objects. The user needs to place its objects on the workspace according to the two views. The top view is also shown on the screen and it is based on the current state of the physical workspace. When the user moves objects on the workspace, the top view is updated but not the two other views since they represent the target position. Such a challenge is illustrated in Figure 4.1. Once all objects are correctly placed the challenge is over, and the time the user took to complete the challenge is recorded.

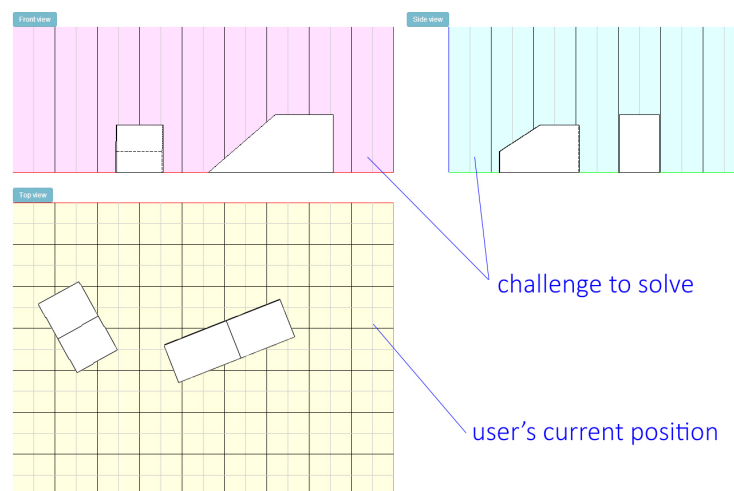


Figure 4.1: Illustration of a challenge.

During the tutorial there is a practice phase. The users have to solve challenges as described before. The difference between the practice phase and the competition phase is that in the practice phase the user is receiving feedback: for each object that users have to place, they can see whether the object is detected and whether the rotation and position are correct. The practice challenges are generated randomly.

During the tutorial, the users also need to create three challenges: one challenge with one object, one with two objects and, finally, one with three objects. These challenges will be solved by the other users during the contest. The more users there will be, the more challenges there are.

Once a user has finished the tutorial he can start solving the other users' challenges to take part of the competition. The user can also go back to the practice part of the tutorial to improve his skills before completing challenges for the competition.

4.1.1 *Security issues*

While testing the challenges we noticed that there are a lot of possibilities to cheat. One issue is that if users could take a challenge more than once, they could complete the challenge and retake it without moving the objects and thus finishing it in no time without moving the objects. To avoid that, we do not allow the users to take a challenge more than once.

Another issue is due to JavaScript: because the code is running on the client side, it is visible to any user. Users knowing that could try to read the code and understand how it works and then find a way to cheat. In the JavaScript code, when a user has completed a challenge, it sends the result (the time the user took to complete the challenge) to the server, which records it. A malicious user, with JavaScript programming skills, could send its result manually to the server with the time he wants. To solve this issue, we changed the code so that it needs to send a hash value (depending on the time, the user ID and the challenge ID) to the server and also the solution of the objects placement. The server needs to verify that everything is valid to accept this result. We also minimized and obfuscated the JavaScript code. This solution is not completely secure but we hope it puts enough obstacles to discourage the majority of the cheaters.

These are the two main issues we noticed and tried to solve as best as possible. Security in JavaScript is problematic since the code is visible to the users. We would have needed more time to find really secure solutions. Also if a user registers with multiple accounts he could use one account to see the challenges and another one to solve them quickly knowing the challenges a priori. It would be hard to find a way to know if two accounts are used by the same person or even by two users cooperating to cheat.

4.2 STUDY SET-UP

The study has been run during the same period as the contest with the challenges. It was from the 21st of May 2013 until the 14th of June 2013. During this period, we gathered data through log files and, at the end of the study, users' feedback from a questionnaire. To recruit partici-

pants, we sent e-mails to more than a hundred people, friends or professional contacts. We also asked colleagues in the CHILI laboratory to advertise the website by e-mails or social network platforms. We told the people that we contacted about the web application and that they could try it if they were interested. We did not want that the user felt pressure, but to attract more people, we offered prizes for the best three users at the end of the contest to attract more people. The prizes were three gift certificates for Amazon, 100\$ for the first place, 50\$ for the second and 20\$ for the third. The study lasted for 25 days and on the 15th day we also posted a news on Hacker News¹, a news aggregator where people can post a URL to “anything that gratifies one’s intellectual curiosity”.

When users accessed the webpage, they were only able to see the homepage. If they wanted to see more and start the tutorial, they needed to subscribe by going through a login protocol. Once they completed the tutorial they had access to the challenges to take part in the contest. Therefore, we distinguished three kinds of users: the visitors, who did not log into the website, the subscribers, who logged in but did not complete the tutorial, and the contestants, who completed the tutorial and took part to the contest. The logs were recorded on the database for different actions made by the subscribers. We recorded on which page they were, and on the different buttons they clicked. We also used Google Analytics² to get several statistics on the website. The advantage of Google Analytics is that it logs information from users who did not subscribe to the online course. Towards the end of the study, we sent questionnaires to the subscribers. There were two different questionnaires, one for the contestants and one for the users who only subscribed to the course. Those questionnaires are shown in the Appendices A and B.

Before our study, we found that a similar research had already been made on TUIs for a larger scale. This is the work of Costanza et al. [7], we presented in Section 2.1. The system was also using a webcam and a workspace and objects with fiducial markers built by the user. They were very optimistic in “bringing TUI to the masses” but did not explain why out of more than 25000 visitors, only 273 managed to use the interface. Those successful users represent about 1% of the total visitors. Costanza just mentioned that it was probably due to incompatible hardware. So one part of the study was trying to understand this behaviour toward home-made TUIs. Another part was to see how usable was this interface and how the users performed.

4.3 RESULTS

The results are separated in three parts. We will first have an overview of all the visitors that went to our website during the study. Then we will focus on users who registered to the course and finally we will analyse the results of the contestants but also the challenges themselves.

4.3.1 *Overview of the visitors*

At the end of the study, Google Analytics recorded 340 unique visitors during 25 days. Out of those visitors, 48 subscribed to the course and only 6 of them completed the tutorial. This is similar to the results of Costanza et al., but on a smaller scale. The advertising with e-mails,

¹ <https://news.ycombinator.com/>

² <http://www.google.com/analytics/>

social networks and Hacker News seemed to have worked well and, as shown in Figure 4.2, the information has been spread worldwide.

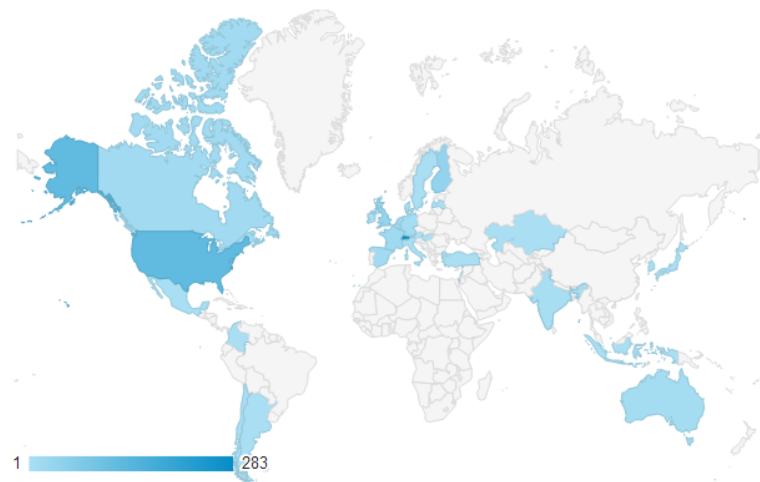


Figure 4.2: Number of visits by country.

But other information on Google Analytics shows that the amount of visitors quickly decreased after the news has been sent. In Figure 4.3, we can see two peaks, the first one was after the initial advertisement with e-mails and the second one after the post on Hacker News. The small peak around the 27th of May was after we asked our colleagues in the laboratory to spread the news to their contacts. Apparently only a few people came back to the website. The log entries for the subscribers confirm this because only the contestants returned to the website, all other subscribers just had a look once.

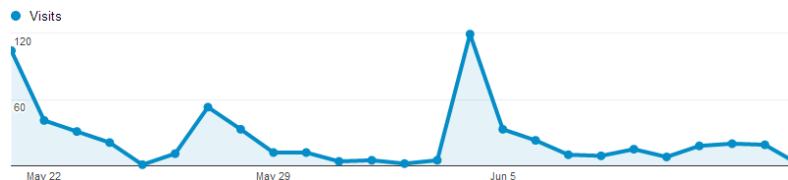


Figure 4.3: Visitors per day.

Looking at when the users left the website gives a better understanding of the patterns of the visit. Figure 4.4 shows that already 57% of the visitors left the website after having seen only the homepage. This means that they did probably not have much interest in 3D geometry or in TUIs. Visitors who tried to see more of the website, needed to log in using a Google, Yahoo, Flickr or AOL account. This login phase discouraged many visitors: 35% of all visitors abandoned at this step, which represents 81% of users who reached the login page. Users may not want to go through a login protocol to see more or maybe they did not have an account from one of the four providers. Finally, only 8% of the visitors reached the first step of the tutorial and there was only 3% who were not directly discouraged by the page describing how to build the TUI.

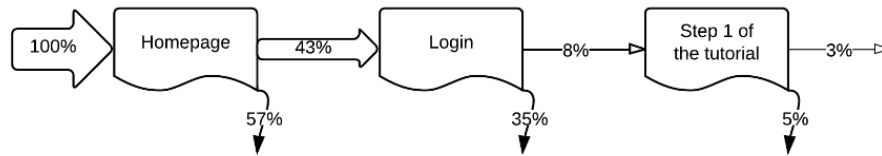


Figure 4.4: Flow of visitors.

Again we see a similar behavior from the users as with the study of Costanza et al. They argued that hardware incompatibility could be a reason why so few visitors really used the system. But in our case, 92% of the visitors did not reach the page explaining the hardware requirements, so the main issue is probably elsewhere. Figure 4.5 shows that 82% of the visitors had a compatible browser (Google Chrome or Mozilla Firefox). So the message at the homepage, suggesting the visitor to use a different browser, may have discouraged at most 18% of the visitors. But we can see that even before the login phase, more than half of the visitors left the website and this shows that it is hard to gain a user's interest. On the first day where there were 79 unique visitors, the average time they spent on the website was 57 seconds. For a web application where users have to spend at least 30 minutes for building their interface, this is very low. We can also imagine that people receiving the e-mail at work could simply not do this directly and then forgot or were not interested enough to do it during their free time. It may be that the potential attraction caused by the originality of home-made TUIs approach is counter-balanced by the effort and time needed, but also by the specific topic that may not interest many users.

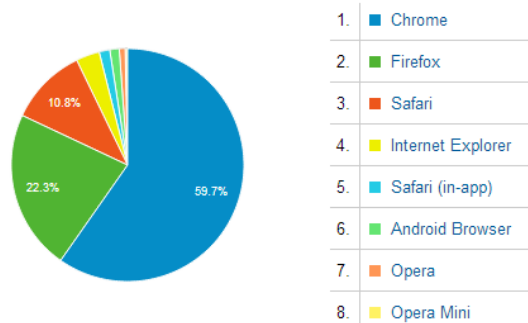


Figure 4.5: Browser market shares.

4.3.2 *Subscribers and the navigation pattern*

If we take a look at the visitors who took time to subscribe, only a few were active. In Figure 4.6, we can see the number of log entries that the database stored for each users. The users on the right side of the vertical line are the contestants and the others were only subscribers. There are four subscribers who were significantly more active. These users had a closer look at the tutorial but did not manage to make it work. One of them, who tried the tutorial in our laboratory but without our help, was not able to understand how to make it work. The reasons were because

of some bugs on Mozilla Firefox with the webcam detection but also because of a lack of details on how to understand the orthographic projections.

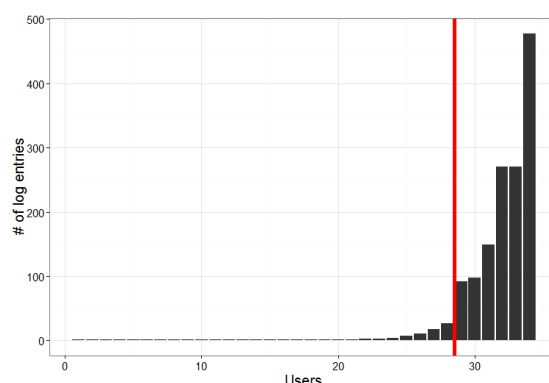


Figure 4.6: Log entries per user.

During the pilot study, some users had trouble navigating through the tutorial. So we stored log entries on the visited pages and Figure 4.7 shows the navigation flow of users for the tutorial during the study. The tutorial workflow should follow this path: first line from left to right, second line from right to left, third line from left to right and then the last page on the fourth line. During the first phase, corresponding to the first line and where users needed to build their TUI, people had to go back and forth in order to complete it. Second page on the first line is the first page where the browser needs access to the webcam, users may have encountered issues to make their webcam work properly and thus tried to refresh the page several times. After the set up phase, users seemed to have followed the workflow as expected. There are two arrows outside of the normal path of the tutorial. The first one represents people arriving directly to the practice page with one object. This was expected because we added a button on the homepage for users who completed the tutorial. The contestants going to the homepage were presented two buttons: one to go to the competition and one to practice. If they clicked on the practice button they were redirected to the practice page from the tutorial. The second arrow not respecting the expected workflow is going from the first step of the third phase to the first step of the fourth and last phase. This is also due to users clicking the practice button, then following the tutorial workflow until they arrive to the third step to create challenges, but since they have already created challenges they go to the “solve challenges” step directly.

With everything users can access through the Internet, they are more difficult to attract. Having users spend 20 minutes on folding papers for a specific topic like 3D geometry is difficult. From the 6 subscribers (but not contestants) who took time to answer the questionnaire, only one said that she did not go through the complete tutorial because of a webcam issue. The five others said that they either did not have time or that it seemed like too much effort. So our web application is not a good solution to attract any users over the Internet. So we now focus on the results of the six contestants.

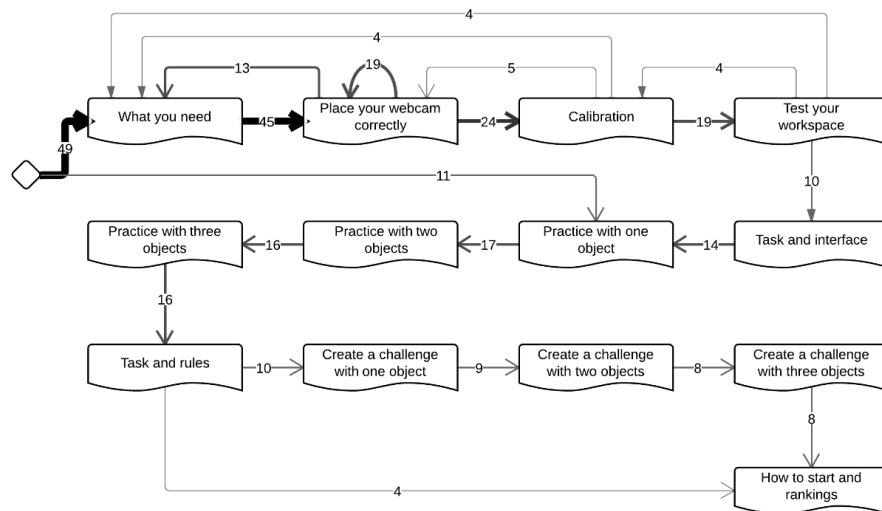


Figure 4.7: Users navigation flow for the tutorial.

4.3.3 Contestants results

Six users took part in the contest. Each contestant had to create three challenges (one with one object, one with two objects and one with three objects) during the tutorial and 6 challenges were already in the database before any user subscribed to the course. Table 4.1 shows some statistics for the challenges. Each contestant could participate in 21 challenges out of the 24 (they could not participate to the challenges they created themselves). Participants tried in average 60.3% of the available challenges, only one contestant completed all the available challenges. More challenges with one object were started, which we interpret as an inclination to solve easier challenges. By looking at the average time for challenge completion, it appears that the time is not proportional to the number of objects. It took more than three times as long to complete a challenge with two objects than a challenge with one object. But the difference between challenges with two objects and challenges with three objects is not as large. So there was a gap for the average time between challenges with one object and challenges with multiple objects. But the average time between challenges with two objects and challenges with three objects is proportional to the number of objects. According to the average time and the percentage of withdrawals, challenges with one object seems to be much simpler. The difference between the difficulty of challenges with two objects and the ones with three objects is harder to determine.

In Figure 4.8, we can see that the values for challenges with one object are less spread and are much lower than for the rest of the challenges. This confirms that they are probably easier to solve. The variance between challenges with three objects is significantly bigger. This is why the box representing the interquartile range for challenges with three objects is larger. So either the difficulty of these challenges can vary a lot or maybe with more data we would have noticed a clearer trend. But with these actual results, it seems again that the difference of difficulty between challenges with two objects and the ones with three objects is not as large as the ones between one object and the other challenges.

# of objects	1	2	3	Total
# of challenges	8	8	8	24
# of challenges taken	29	24	23	76
% of participation	69	57.1	54.8	60.3
Average time in seconds	40	132	185	112
% of withdrawals	0	12.5	17.4	9.2

Table 4.1: Statistics on challenges.

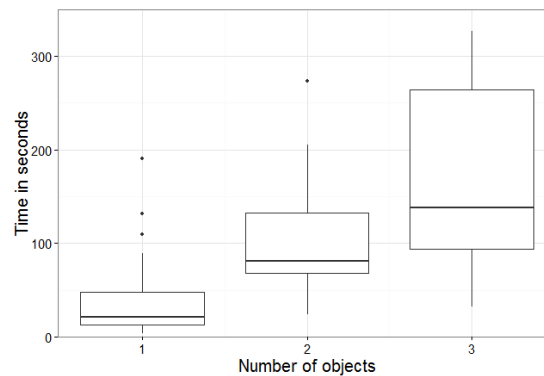


Figure 4.8: Time taken for challenges with 1, 2 or 3 objects.

If we look at the average time by user given the number of objects in the challenges (Figure 4.9), we cannot really draw general conclusions. For example User1 slope is quite linear, while User6's slope is looking like an exponential and User4's slope is more logarithmic. But only User2 has a smaller average time for challenges with three objects than challenges with two objects, so we took a closer look at his results.

He was the youngest user (13 years old) and he attempted all challenges. He first tried all the challenges with one object then the challenges with two objects and finally the challenges with three objects. He was active on three different days. On the first day he completed two challenges with one object. The next day he started taking practice challenges, then completed all the other available challenges with one object. Before taking challenges with two objects he went back to the practice phase and only after that attempted three challenges with two objects. Out of these three challenges he did not manage to finish the two first. He left the website and came back three days later and again started with the practice phase. He then completed all the available challenges left and was not able to complete two of the challenges with three objects. We hope this TUI helped him improve his spatial skills and from his results on the average time taken for challenges it is a possibility. But it may also be possible that he was handling the TUI with more ease on the third day. He showed a lot of interest with this competition, checking the ranking page frequently during and after he finished all the available challenges. We were

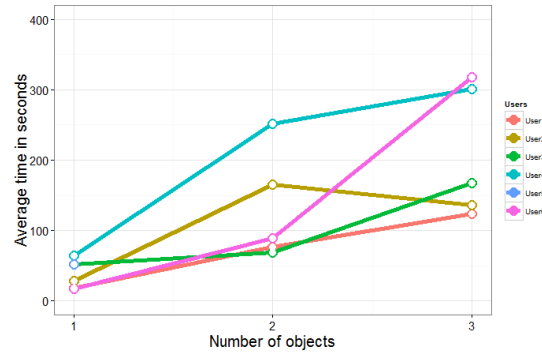


Figure 4.9: Users' average time for challenges with 1, 2 or 3 objects.

also glad to see that he tried to practice before taking challenges, which is a behavior we were expecting for someone who wants to improve his skills. One other user practiced once before taking challenges. So he is the only one who practiced more regularly.

From the statistics above, it may be plausible that at least one user was improving. But when looking at Figure 4.10, the time needed to complete the challenges with one object is not necessarily decreasing over time. The same is happening for challenges with two or three objects. Apparently challenges from the same number of objects can be more or less difficult. This might mean that some objects are easier to understand than others.

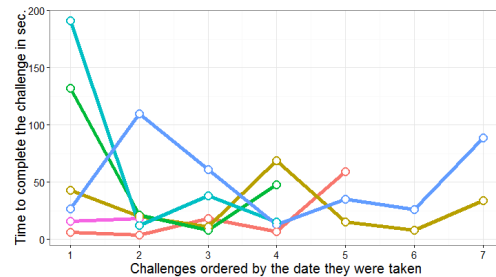


Figure 4.10: Time to complete a challenge with one object.

4.3.3.1 *Difference between the three objects*

Table 4.2 shows some statistics depending on the objects in a challenge. For challenges with one object, the second object gave more difficulties to the users than other objects. This is confirmed with challenges with two objects where challenges containing the second object took in average more time than the challenges without this object. Moreover, the only challenges that were not completed by the users were the ones containing this object.

This object has two faces with different slopes but in Figure 4.11 it is hard to distinguish those two slopes with orthographic projections. The user can easily be wrong by 180° on the rotation of the object. Cuendet et al. have already noticed that objects with a higher degree of symmetry









								
Object								
# challenges	3	4	1	3	2	3		8
# results	9	19	1	13	2	9		23
# of withdrawals	0	0	0	2	0	1		4
Avg. time (in sec.)	15	52	26	126	90	151		185

Table 4.2: Statistics for challenges by objects.

are more difficult to understand on orthographic projections [8]. Although the degree of symmetry of this object is identical to that of the other objects, it is closer to having a second axis of symmetry than the other two objects.

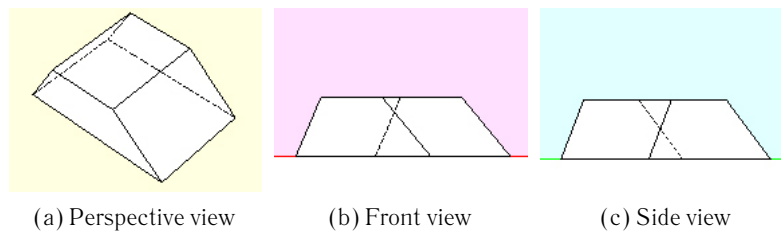


Figure 4.11: A specific orientation that makes this object difficult to understand.

4.3.3.2 Rendering plain objects

To help users understand the views they are seeing, we added a button to switch between transparent objects and plain objects. This can be faster to understand the depth information than with stippled lines only. Figure 4.12 shows how often contestants were using this button. The users in the graphs are ordered given the final ranking of the contest. We can see that, except the third user who never used it, the better a user is ranked, the more he used the button. The third user had probably less difficulties than other users to interpret the stippled lines.

In Figure 4.13, we can see that users were clicking more on the button for challenges with three objects. When there are more objects, there are also more often stippled lines because the workspace is not very big, so objects are superimposed on the orthographic views. Displaying objects without the transparency helps the user to understand faster which object is in front and which one is behind without any additional interpretation in his mind. Before the study we were not sure if users would use this button or not, but it seems that it was often used and also useful for some contestants.

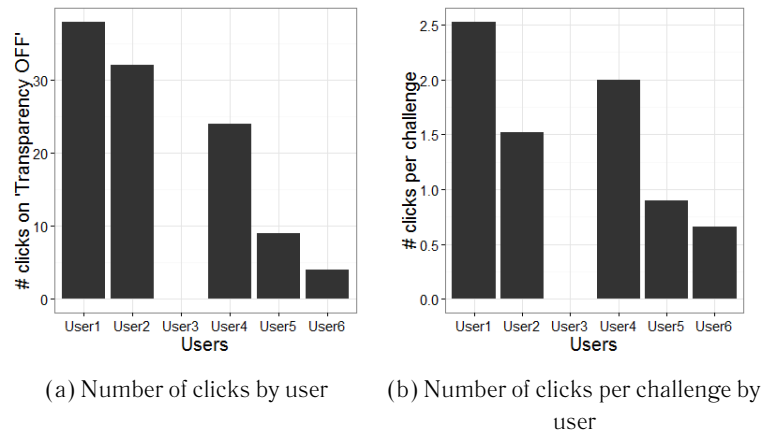


Figure 4.12: Statistics by user on the "Transparency OFF" button.

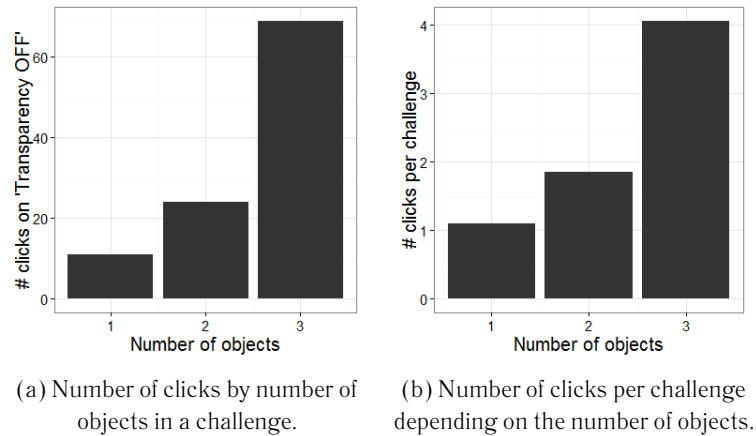


Figure 4.13: Statistics on the "Transparency OFF" button by number of objects in a challenge.

4.3.3.3 Questionnaire

From the six contestants, five of them answered the questionnaire in Appendix B. We first asked some personal question about them (sex, age, level of education and domain). There was a 13 years old boy still attending mandatory school, a man (over 46 years old) working in insurance, two IT Ph.D. between 26 and 35 years old and a woman between 26 and 35 years old with a Master in IT. Those results might be biased because of the three users with an IT background.

TUTORIAL

The first topic of the questionnaire was the tutorial, for which we were wondering how difficult, intuitive, and frustrating it would be. The difficulty of the tutorial, illustrated in Figure 4.14, depends on the step. Creating the challenges seemed to be the easiest step, people were just asked to put their objects on the workspace and then click a button to save the challenge in the database. The feedback on the practice part is also mostly positive. The creation of the objects and the workspace was not too difficult either although it takes time. The only step in the tutorial

that was hard for one person was placing the webcam, this was the older user who also does not have a background in IT. Even though this issue has been improved after the pilot study, this may still need some improvements.

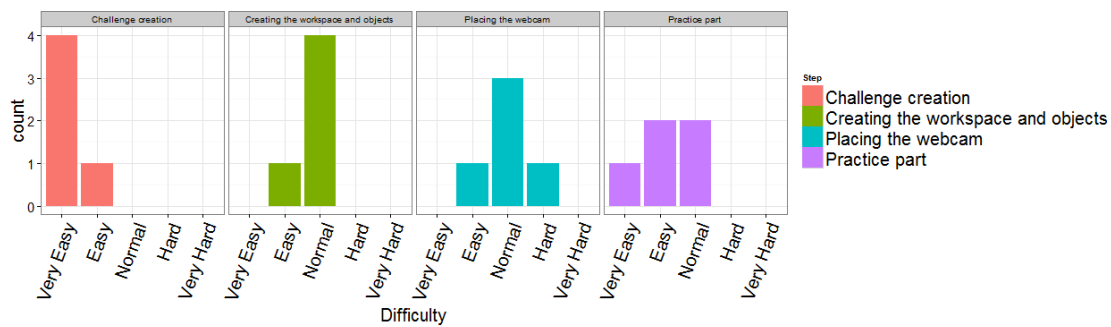


Figure 4.14: Answers about the difficulty of the tutorial.

The second question about the tutorial was about the intuitiveness of its steps. The results in Figure 4.15 shows that two steps were not very intuitive: placing the webcam and creating the workspace and objects. The creation of the workspace and the objects needed more material (cutter or scissor, glue and tape). Building shapes, they have never seen before, is not trivial and there were only pictures, no video to help them. So those results seem logical. And again the tutorial for placing the webcam is not really convincing which is coherent with the previous result.

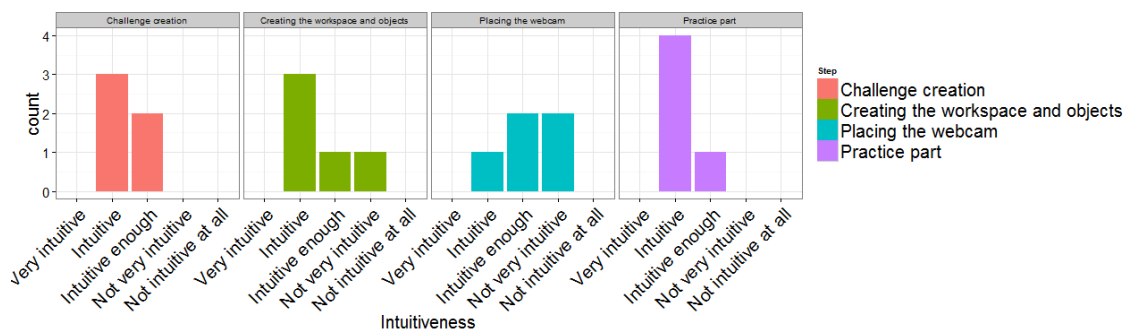


Figure 4.15: Answers about the intuitiveness of the tutorial.

The third question about the tutorial was to enquire about the level of frustration of the user (see Figure 4.16). Similarly to the intuitiveness, placing the webcam and creating the workspace and objects recorded some bad feedback. But only one user was frustrated by the creation of the interface. This is a bit surprising because most of the visitors probably left the website because it was too much effort, so we were expecting more frustration from this part. Placing the webcam was frustrating for two users and, as said before, needs to be redesigned to achieve better performance.

The four steps in the previous questions can be separated in two groups: set up steps and use steps. The steps that received bad feedback were only from the set up group. Apparently using

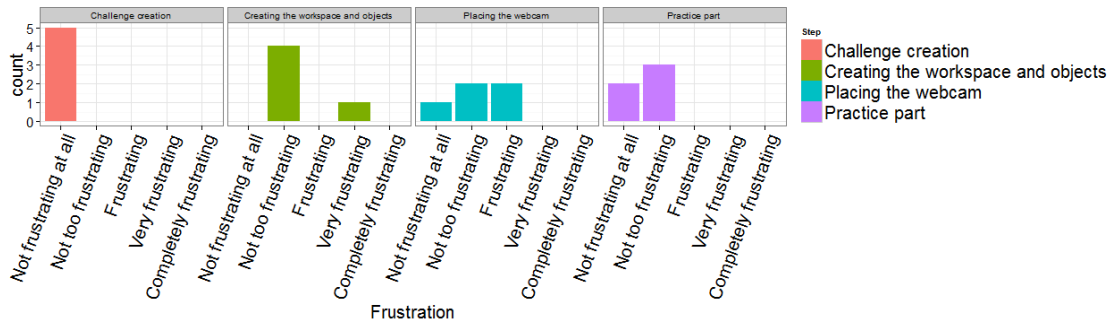


Figure 4.16: Answers about the frustration coming from the tutorial.

the TUI afterwards does not lead to problems. This shows again that users can interact easily with a TUI.

CHALLENGES

The next part of the questionnaire focused on the challenges. From the answers, they all found the challenges fun and interesting. The difficulty was adequate for three users and difficult for the two others. They all thought that this activity of linking 2D and 3D representations can be useful even though only the 13 years old boy claimed to be interested in developing his 3D skills. At the end, the challenges received only positive feedback from this questionnaire.

INTERFACE

The last part of the questionnaire was focused on the TUI itself. First we wanted to know if seeing the physical objects was helping them to solve the challenges. Three of them said that it helped them, one did not think so and the last one was not sure. The one, who did not find the objects helpful to solve the challenges, was probably able to internally represent the 3D representation out of the orthographic views. We were then wondering if they thought that having tokens instead of real 3D objects is important or not. Four of the participants said it was really helpful to have the 3D objects and only one claimed he would just have been less efficient and would have gained a lot of time during the building phase. This is the same user who found the creation of objects very frustrating. Another interesting question was, what they were looking at while solving a challenge. There are two kinds of users, the ones who look at the screen and the workspace alternatively and the ones who look at the screen without looking often at the workspace. It's like a keyboard, once people get used to it, they may not need to look at it anymore. And two, out of the five users, were looking at the screen without looking at the workspace too often. One of the two is the one who did not think that having the real physical objects is helpful to solve the challenges.

The users also had the possibility to describe problems they may have encountered with the interface. Someone said they would have liked to have a smaller workspace because he had to put his webcam too high to calibrate properly. This is actually possible if the user prints every PDF file with the same lower ratio. This might be something we can add in the tutorial. One user said that the system crashed once and another that it did not work on all of his computers. He was using Linux so it was probably an issue with the graphic card drivers.

4.4 DISCUSSION

The results of this work were mixed. A very positive aspect was that we managed to have a self-explanatory website and a home-made TUI usable for anyone who meets the minimal hardware and browser requirements. We were able to fix the minor usability issues early on and, during the deployment, the system worked well. We received positive unsolicited comments from two contestants: “Very cool project. Worked surprisingly well. Good luck!” and “There is a big investment for the building phase. But it’s really worth it, it’s really great. [...] The challenges are addictive with the points and ranking [...] I was impressed how well the TUI works”. So we achieved our main goal, which was to build a tangible interface that could be easily scaled up, with a minimum hardware and software requirements.

A more mixed result was the participation to the study. The rather small number of participants indicates that even though it is easy nowadays to advertise an online application all over the world, it is also hard to get users to commit a significant amount of time for an application that is not recreational or at the core of their interest. It may be that if we had tested the same applications with an audience in need of spatial skills training, we could have had a higher rate of participation, and it is definitely something that we would like to explore in the future. The upfront time investment needed to build the interface was rather large, and reducing the investment time needed could lower the barrier for participation. There are several solutions to shorten the building phase. We could give the option for users to use flat representations of objects, so they would only need to cut the shape in paper, no folding and gluing. We could also use standard blocks like the Lego bricks to construct the different tangible objects. 3D printers will probably be more affordable in the future and this would be an easy alternative to create the objects. For a professional school for carpenters, the teacher could easily cut many of these objects out of wood, and then provide them to the students. Even if the process was longer, using it for a whole school year could make the effort worth it. Another option to decrease the time investment could be to design applications for other domains, where the building of the interface would not be as complex. For example, Bonnard et al. have done activities to develop 2D geometry skills using the TinkerLamp [5], where the objects of the interface were in 2D shapes and could be simply cut out of paper.

We received positive feedback from the contestants about the challenges. These users were maybe attracted by the introductory video, the prizes, or by TUIs in general, which is why they took time to complete the tutorial and gave positive feedback. If we had forced users to use the system, it may be that their feedback would have been less positive. Costanza et al. were optimistic after their study because they received mostly positive feedback. But this is probably because users were interested even before trying the TUI. Building the interface is filtering the less interested users and then biases the sample of users because people who do not like this TUI simply will not try it. Costanza et al.’s TUI is not a learning TUI, it is only made for entertainment, but our TUI has a specific purpose and if it is used in schools, the context is not the same, so it may well be that the number of users would remain constant over time.

From a pedagogical point of view, we noticed that the 13 years old boy, who claimed to be interested in developing his 3D skills, was the one who was the most likely to have improved

his spatial skills according to our data. If, with further research, we can show that this TUI is improving the users' spatial skills, this TUI may start to be used in a real context and hopefully have many frequent users. Since the challenges are very similar to one activity from TapaCarp we can hope that the online version is indeed helping users with their 3D skills. But without a dedicated study it remains just an hypothesis. Cuendet et al. [8] showed that the degree of symmetry increases the difficulty to understand the orthographic projections and we were able to observe the same phenomenon in our study. In the questionnaire the contestants said that they would be less efficient if they had only tokens instead of the 3D physical objects. Of course, this is just a self-perceived judgement from the participants themselves, but this coincides with the results of Cuendet et al. [8] where users were less efficient with the use of tokens.

CONCLUSIONS

5.1 CONCLUSIONS AND FUTURE WORK

The goal of the project to create an alternative low-cost and online version of TapaCarp is achieved. We managed to use only HTML5 standards to access the webcam and render 3D graphics. The tangible interface is working well, it is reactive, fluid, and its usability is good. The contest was appreciated by the participants and was even found addictive for competitive people. After the pilot study, there were also people visiting our laboratory at EPFL who showed some interest for the possible impact on learning of a TUI like ours.

There are many factors and constraints to consider when scaling up a TUI. We tried to make our system as low in requirements as possible: beside a computer, only a cheap webcam and a regular printer are needed. An internet connection (to access a server) is only needed for the calibration phase. While a user is taking a practice challenge, everything is done on the client side, and the user can even generate a new challenge on the same page without requesting any connection to the server. This can be an important factor for developing countries, which generally have low bandwidth. Also we used Google Course Builder, an open source project to implement MOOCs, and these online courses are becoming more and more popular. This is thus interesting to have a solution mixing online education and TUI for learning.

We did two studies, after the first one we were able to see how a user would interact with the system. We observed what was clear during the tutorial and what needed more improvements. Thanks to these improvements the second study received positive feedback and users did not seem to struggle much to build their interface. Even with the few information we gathered from the contestants, we were able to notice some interesting results, as shown in [Section 4.3](#).

There were also things that were not working well. We were not able to attract many active users. The installation of the webcam was more difficult for users than what we expected. The long process to create the objects out of paper can be frustrating or discouraging for many users. Fortunately if we expect the target users to be schools or organizations in the educational field, there are several options to avoid these problems and mainly the time consuming building phase. As explained in [Section 4.4](#) we could, for example, let the users choose between using 3D objects or their flat representation. One major issue though is that we cannot say yet if this TUI is helping the users to improve their spatial skills. And of course if we had more users and especially more users without a background in IT, our results could have been more interesting.

This work was just a starting point to see if an online version of TapaCarp was feasible with the increase of HTML5 standards and all the open source JavaScript libraries available. The positive outcome of this project can lead to a lot of improvements and other work with it.

We need now to do some research on the learning impact of this TUI for its users. If the impact on learning of this TUI is positive, it can have a promising future. Also it needs to have more activities, like TapaCarp provides. But other activities for different target skills can be imagined or also reused from the different TinkerLamp applications developed in the CHILI laboratory. So knowing if this TUI can improve the learning process of users for some skills and developing more activities should be a priority.

We can also imagine doing some studies on the difference between the online version of TapaCarp and the one using the TinkerLamp. Discrete mode of representation against the co-located mode, is there a significant difference for the learning outcome? Do users have a preference between the two versions? Do users perform better with the “Transparency OFF” button? If they improve their skills, does this button also have an impact on their improvement ?

Again if this TUI has been proved useful for learning purposes, another future focus can be to make some advertisement to any educational organization that might benefit from it. Making this TUI a useful product and not just a interesting work from a research point of view can be a great achievement.

As mentioned before, this is only a first step. We strongly believe in the potential of TUIs for learning, and we think that the new technological advances offer a great opportunity to scale them up and democratize their use. At a time where MOOCs are revolutionizing higher education, it would be interesting to think of ways in which a TUI such as this one could be used in a MOOC.

QUESTIONNAIRE FOR SUBSCRIBERS

Questionnaire for tangiblecourse.appspot.com

Data will be anonymised and used for research purposes only.

* Required

Gender *

- ☐ Male
☐ Female

Age *

- ☐ 16 - 25
☐ 26 - 35
☐ 36 - 45
☐ 46 - more

Level of education *

- ☐ Bachelor
☐ Master
☐ PhD
☐ Other:

Domain *

- ☐ Carpentry
☐ IT
☐ Engineering
☐ Other:

I completed the following steps: *

- ☐ Watched the introductory video
☐ Printed the PDF files
☐ Folded the paper blocks
☐ Set up my webcam
☐ Tried the practice challenges
☐ Other:

How much time did you spend doing the previous steps? *

- ☐ less than 10 minutes
- ☐ between 10 and 20 minutes
- ☐ between 20 and 30 minutes
- ☐ more than 30 minutes

Why didn't you try to go further? *

- ☐ This seemed like too much effort
- ☐ I did not have an external webcam
- ☐ I did not have access to a printer
- ☐ I did not have enough time
- ☐ I do not have any interest for 3D geometry
- ☐ I was frustrated
- ☐ Other:

Do you think tangible interfaces could be used for other domains than 3D geometry? If so, which ones?

Do you have any suggestions or feedback?

Critics, ideas, bugs, congratulations, etc.

QUESTIONNAIRE FOR CONTESTANTS

Questionnaire for tangiblecourse.appspot.com

Data will be anonymised and used for research purposes only.

* Required

Gender *

- ☐ Male
☐ Female

Age *

- ☐ 16 - 25
☐ 26 - 35
☐ 36 - 45
☐ 46 - more

Level of education *

- ☐ Bachelor
☐ Master
☐ PhD
☐ Other:

Domain *

- ☐ Carpentry
☐ IT
☐ Engineering
☐ Other:

Questionnaire for tangiblecourse.appspot.com

* Required

Tutorial

How difficult was the tutorial? *

Did you encounter some troubles?

	Very easy	Easy	Normal	Hard	Very hard
Creating the workspace and shapes	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Placing the webcam	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Practice part	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Challenge creation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

How intuitive was the tutorial? *

	Very intuitive	Intuitive	Intuitive enough, but not great	Not very intuitive	Not intuitive at all
Creating the workspace and shapes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Placing the webcam	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Practice part	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Challenge creation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

How frustrating was the tutorial? *

	Not frustrating at all	Not too frustrating	Frustrating	Very frustrating	Completely frustrating
Creating the workspace and shapes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Placing the webcam	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Practice part	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Challenge creation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Activity**How easily were you able to find on the website the information that you needed? ***

1 2 3 4 5

Very easily ☐ ☐ ☐ ☐ ☐ It was very hard**How fun was the activity that you had to perform? ***

1 2 3 4 5

Not at all ☐ ☐ ☐ ☐ ☐ Very**How interesting was the activity? ***

1 2 3 4 5

Not at all ☐ ☐ ☐ ☐ ☐ Very**How easy/difficult was it to solve the challenges? ***

1 2 3 4 5

Easy ☐ ☐ ☐ ☐ ☐ Very difficult

Do you think that an activity that makes you link the 2D and 3D representations of an object is useful? *

1 2 3 4 5

Not useful at all ☐ ☐ ☐ ☐ ☐ Very useful

How interested are you in developing your 3D geometry skills? *

1 2 3 4 5

Not at all interested ☐ ☐ ☐ ☐ ☐ Very interested

Interface

Do you think that having the physical objects helped you solve the challenges? *

1 2 3 4 5

Not at all ☐ ☐ ☐ ☐ ☐ Totally

Do you think a flat representation of the object would have been sufficient? *

By flat representation, we mean a piece of paper having the same shape as the top view of the object

- ☐ Yes, a flat representation would have made no difference at all.
- ☐ I would not have been as efficient, but it would have saved me a lot of time in the phase of building the interface.
- ☐ No, seeing the 3D object was really helpful, a flat representation would have made it harder to solve these kind of problems.
- ☐ I'm not sure, I would have to try.

If you had completed the same activity with just a mouse and keyboard (i.e. no tangible objects), you would have... *

- ☐ ... had more fun
- ☐ ... had the same amount of fun
- ☐ ... had less fun


While solving a challenge, I was looking at ... *

- ☐ ... the shapes and workspace mostly
- ☐ ... the screen mostly
- ☐ ... the screen and objects quite equally
- ☐ I do not remember


Did you have any particular problem regarding the use of the interface? *

A large, empty rectangular text box with a thin border and a small diagonal line in the bottom right corner, intended for the user to describe any problems with the interface.

Did you have any problems using the website (e.g. browser did not work, webcam was not compatible, etc.)? *

A large, empty rectangular text box with a thin border and a small diagonal line in the bottom right corner, intended for the user to describe any problems with the website.

Do you think tangible interfaces could be used for other domains than 3D geometry? If so, which ones?

A large, empty rectangular text box with a thin border and a small diagonal line in the bottom right corner, intended for the user to answer whether tangible interfaces could be used in other domains and specify which ones.

« Back

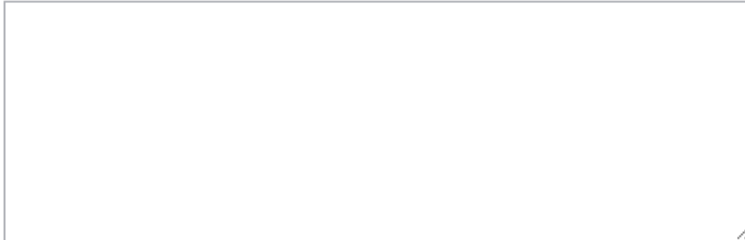
Continue »

Questionnaire for tangiblecourse.appspot.com

Suggestions

Do you have any suggestions or feedback?

Critics, ideas, bugs, congratulations, etc.

A large, empty rectangular box with a thin black border, intended for users to provide suggestions or feedback. A small cursor icon is visible in the bottom right corner of the box.

BIBLIOGRAPHY

- [1] Shaaron Ainsworth. The functions of multiple representations. *Comput. Educ.*, 33(2-3):131–152, September 1999.
- [2] Shaaron Ainsworth. Deft: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, 16(3):183 – 198, 2006.
- [3] David Anderson, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S. Yedidia. Tangible interaction + graphical interpretation: a new approach to 3d modeling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 393–402, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [4] P. Blikstein. Travels in troy with freire: technology as an agent for emancipation. In *Social Justice Education for Teachers: Paulo Freire and the possible dream*, pages 205–244. Sense, 2008.
- [5] Quentin Bonnard, Himanshu Verma, Frédéric Kaplan, and Pierre Dillenbourg. Paper Interfaces for Learning Geometry. In *7th European Conference on Technology Enhanced Learning*, 2012.
- [6] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 423–432, New York, NY, USA, 2008. ACM.
- [7] Enrico Costanza, Matteo Giaccone, Olivier Kueng, Simon Shelley, and Jeffrey Huang. Ubicomp to the masses: a large-scale study of two tangible interfaces for download. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, pages 173–182, New York, NY, USA, 2010. ACM.
- [8] Sébastien Cuendet, Engin Bumbacher, and Pierre Dillenbourg. Tangible vs. virtual representations: when tangibles benefit the training of spatial skills. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, NordiCHI '12, pages 99–108, New York, NY, USA, 2012. ACM.
- [9] Sébastien Cuendet and Pierre Dillenbourg. The Benefits and Limitations of Distributing a Tangible Interface in a Classroom. In *CSCW 2013*, 2013.
- [10] Sébastien Cuendet, Patrick Jermann, Christophe Arn, and Pierre Dillenbourg. A Study of Spatial Reasoning Skills in Carpenters Training. 2011.
- [11] Sébastien Cuendet, Patrick Jermann, and Pierre Dillenbourg. Tangible interfaces: when physical-virtual coupling may be detrimental to learning. In *Proceedings of the 2012 British Computer Society Conference on Human-Computer Interaction*, 2012.

- [12] Son Do-Lenh, Patrick Jermann, Sébastien Cuendet, Guillaume Zufferey, and Pierre Dillenbourg. Task performance vs. learning outcomes: a study of a tangible user interface in the classroom. In *Proceedings of the 5th European conference on Technology enhanced learning conference on Sustaining TEL: from innovation to learning and practice*, EC-TEL'10, pages 78–92, Berlin, Heidelberg, 2010. Springer-Verlag.
- [13] George W. Fitzmaurice and William Buxton. An empirical evaluation of graspable user interfaces: towards specialized, space-multiplexed input. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, CHI '97, pages 43–50, New York, NY, USA, 1997. ACM.
- [14] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. Bricks: laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 442–449, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [15] Michael S. Horn, Erin Treacy Solovey, and Robert J. K. Jacob. Tangible programming and informal science learning: making tuis work for museums. In *Proceedings of the 7th international conference on Interaction design and children*, IDC '08, pages 194–201, New York, NY, USA, 2008. ACM.
- [16] Hiroshi Ishii. The tangible user interface and its evolution. *Commun. ACM*, 51(6):32–36, June 2008.
- [17] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, CHI '97, pages 234–241, New York, NY, USA, 1997. ACM.
- [18] D. Klahr, L. M. Triona, and C. Williams. Hands on what? The relative effectiveness of physical versus virtual materials in an engineering design project by middle school children. *Journal of Research in Science Teaching*, 44:183–203, January 2007.
- [19] Jill H. Larkin and Herbert A. Simon. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science*, 11(1):65–100, 1987.
- [20] Andrew Manches, Claire O'Malley, and Steve Benford. The role of physical representations in solving number problems: A comparison of young children's use of physical and virtual materials. *Comput. Educ.*, 54(3):622–640, April 2010.
- [21] Nicole McNeil and Linda Jarvin. When theories don't add up: Disentangling the manipulatives debate. In *Theory Into Practice*, pages 309–316, Philadelphia, PA, USA, 2007. Lawrence Erlbaum.
- [22] Sara Price. A representation approach to conceptualizing tangible learning environments. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, TEI '08, pages 151–158, New York, NY, USA, 2008. ACM.

- [23] Sara Price, Taciana Pontual Falcão, Jennifer G. Sheridan, and George Roussos. The effect of representation location on interaction in a tangible learning environment. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pages 85–92, New York, NY, USA, 2009. ACM.
- [24] Mitchel Resnick, Fred Martin, Robert Berg, Rick Borovoy, Vanessa Colella, Kwin Kramer, and Brian Silverman. Digital manipulatives: new toys to think with. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 281–287, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
- [25] Orit Shaer and Eva Hornecker. Tangible user interfaces: Past, present, and future directions. *Found. Trends Hum.-Comput. Interact.*, 3(1–2):1–137, January 2010.
- [26] Irving E. Sigel. Developmental trends in the abstraction ability of children. *Child Development*, 24(2):pp. 131–144, 1953.
- [27] L. M. Triona and D. Klahr. Point and click or grab and heft: Comparing the influence of physical and virtual instructional materials on elementary school students' ability to design experiments. In *Cognition and Instruction*, pages 149–173, 2003.
- [28] Brygg Ullmer, Hiroshi Ishii, and Robert J. K. Jacob. Token+constraint systems for tangible interaction with digital information. *ACM Trans. Comput.-Hum. Interact.*, 12(1):81–118, March 2005.
- [29] Bieke Zaman, Vero Vanden Abeele, Panos Markopoulos, and Paul Marshall. Editorial: the evolving field of tangible interaction for children: the challenge of empirical validation. *Personal Ubiquitous Comput.*, 16(4):367–378, April 2012.
- [30] Jiajie Zhang, Jean M, Mark St, and Donald A. Norman. Representations in distributed cognitive tasks. *Cognitive Science*, 18:87–122, 1994.
- [31] Guillaume Zufferey, Patrick Jermann, Aurélien Lucchi, and Pierre Dillenbourg. Tinker-Sheets: Using Paper Forms to Control and Visualize Tangible Simulations. In *TEI'09, Third International Conference on Embedded and Tangible Interaction*.